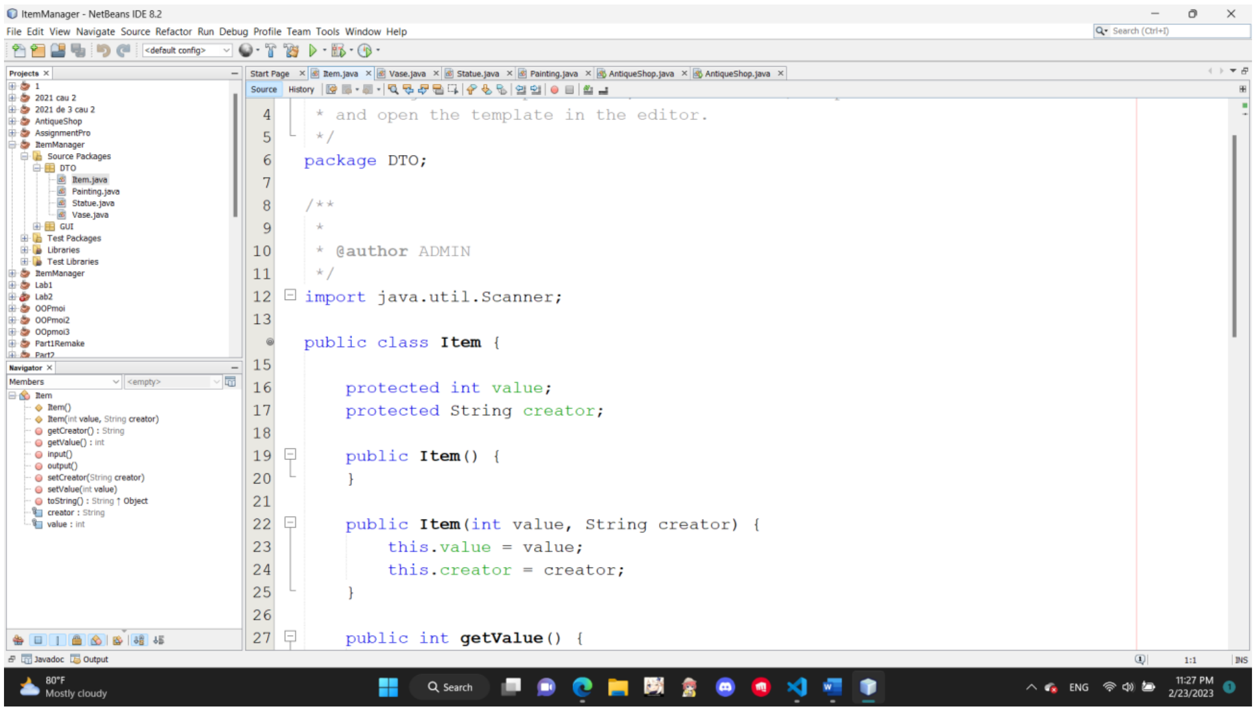
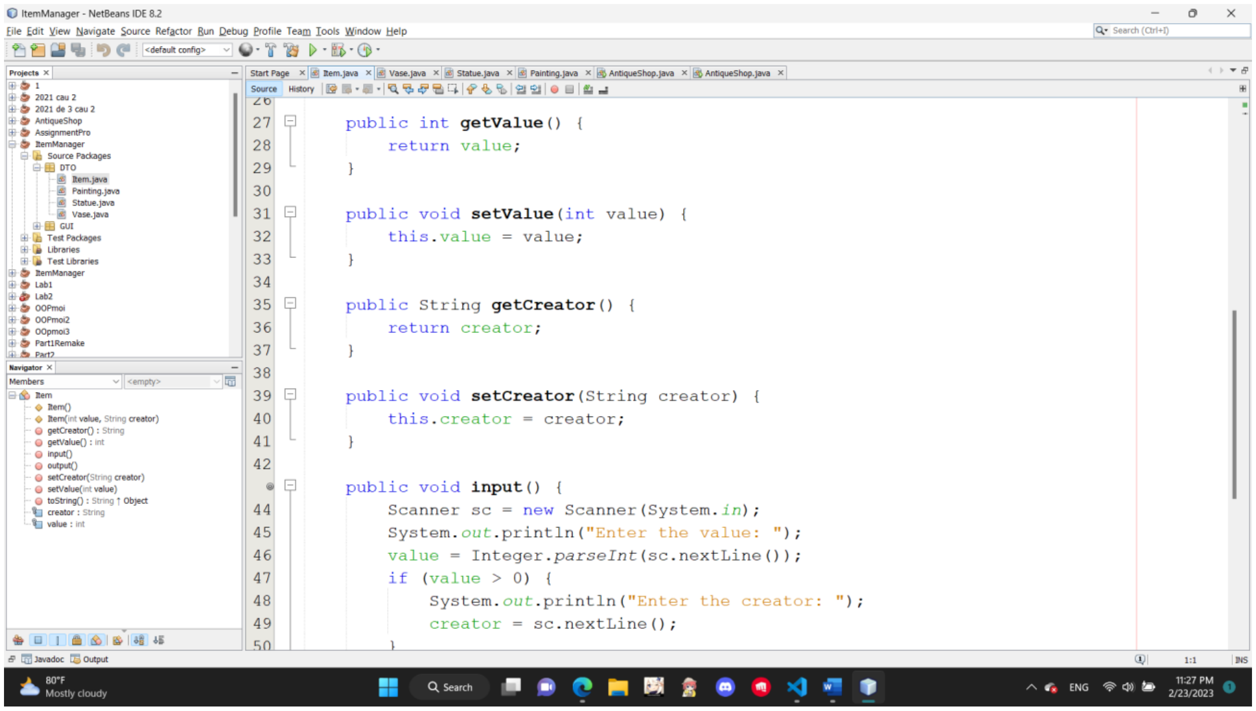
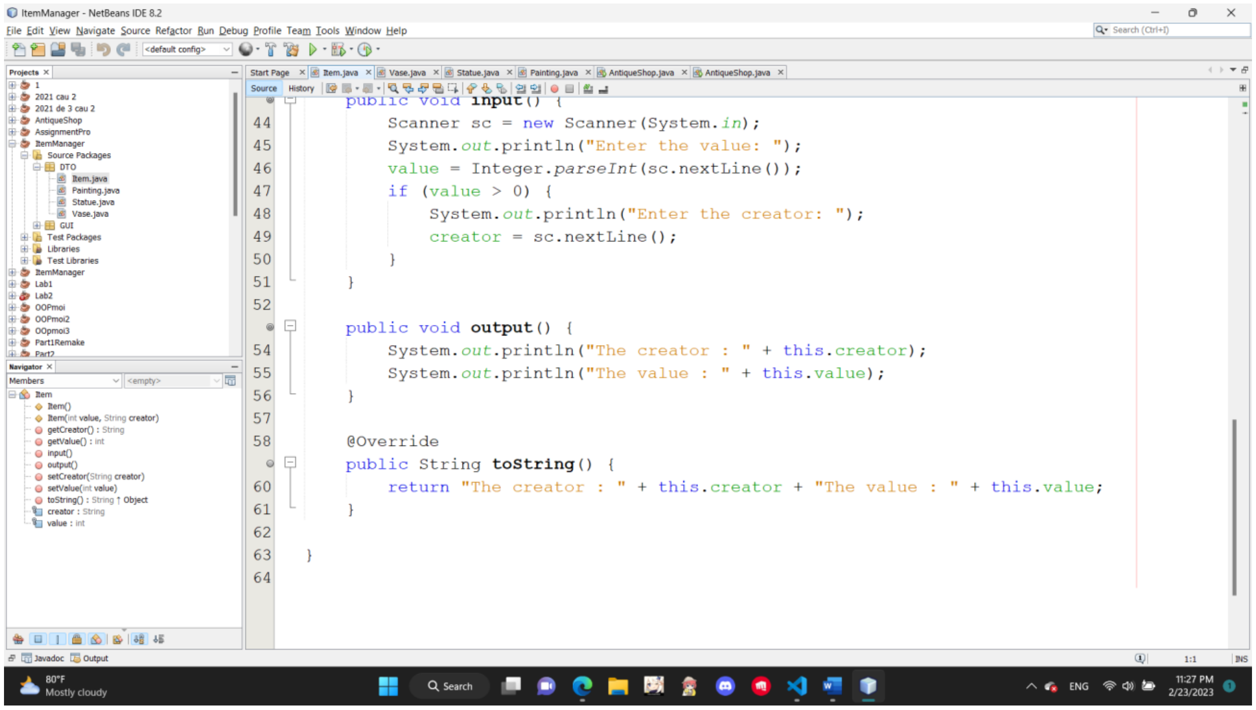
Item:



```java
 * and open the template in the editor.
 */
package DTO;

/**
 *
 * @author ADMIN
 */
import java.util.Scanner;

public class Item {

    protected int value;
    protected String creator;

    public Item() {
    }

    public Item(int value, String creator) {
        this.value = value;
        this.creator = creator;
    }

    public int getValue() {
```

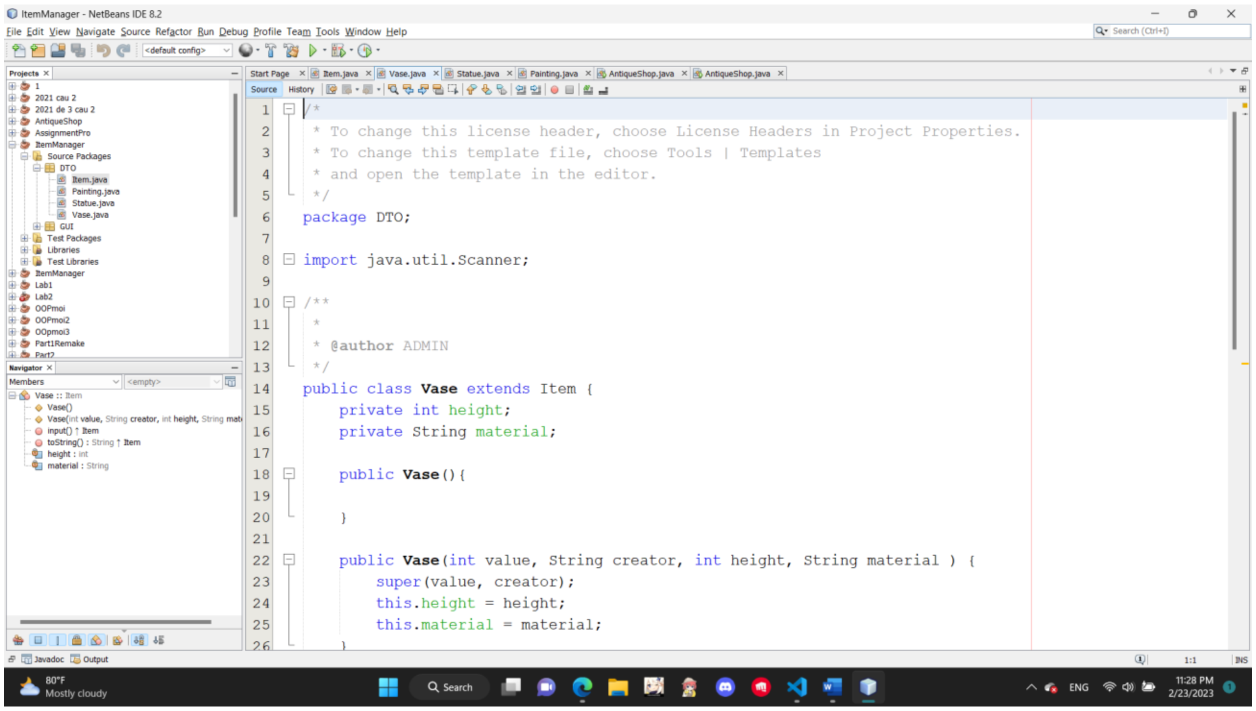


```java
    public int getValue() {
        return value;
    }

    public void setValue(int value) {
        this.value = value;
    }

    public String getCreator() {
        return creator;
    }

    public void setCreator(String creator) {
        this.creator = creator;
    }

    public void input() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the value: ");
        value = Integer.parseInt(sc.nextLine());
        if (value > 0) {
            System.out.println("Enter the creator: ");
            creator = sc.nextLine();
        }
```

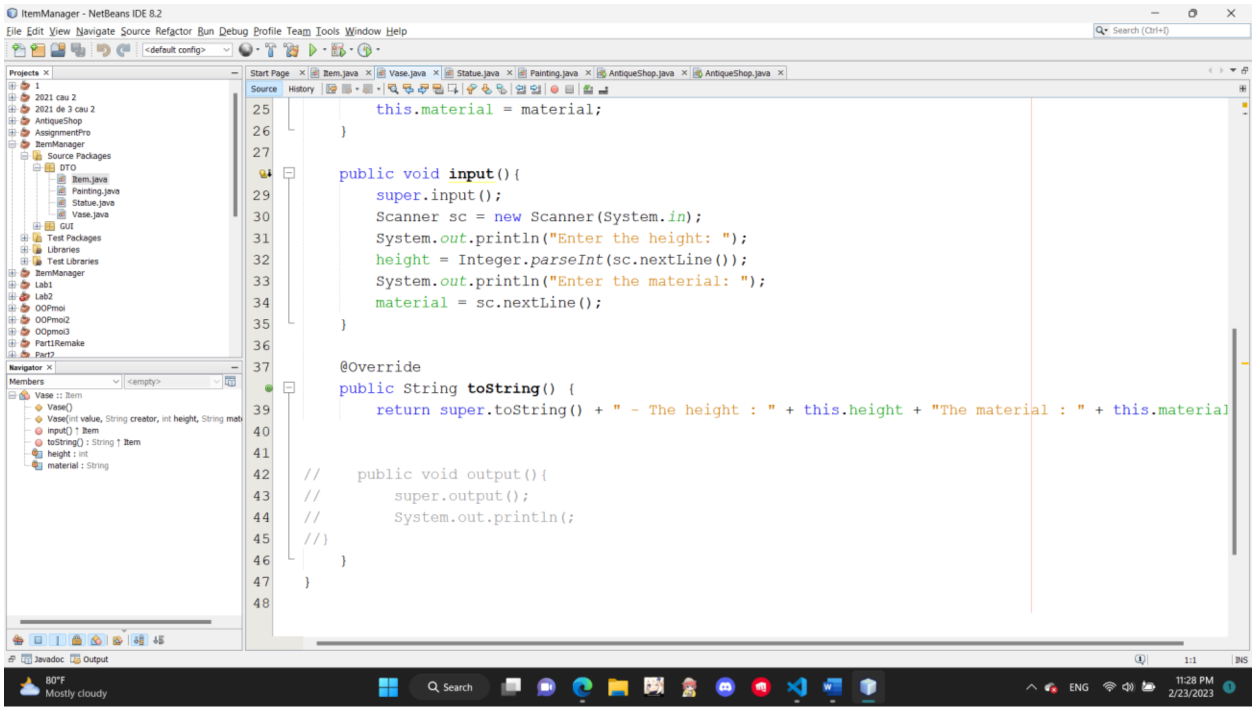


```java
    public void input() {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the value: ");
        value = Integer.parseInt(sc.nextLine());
        if (value > 0) {
            System.out.println("Enter the creator: ");
            creator = sc.nextLine();
        }
    }

    public void output() {
        System.out.println("The creator : " + this.creator);
        System.out.println("The value : " + this.value);
    }

    @Override
    public String toString() {
        return "The creator : " + this.creator + "The value : " + this.value;
    }

}
```

Vase:

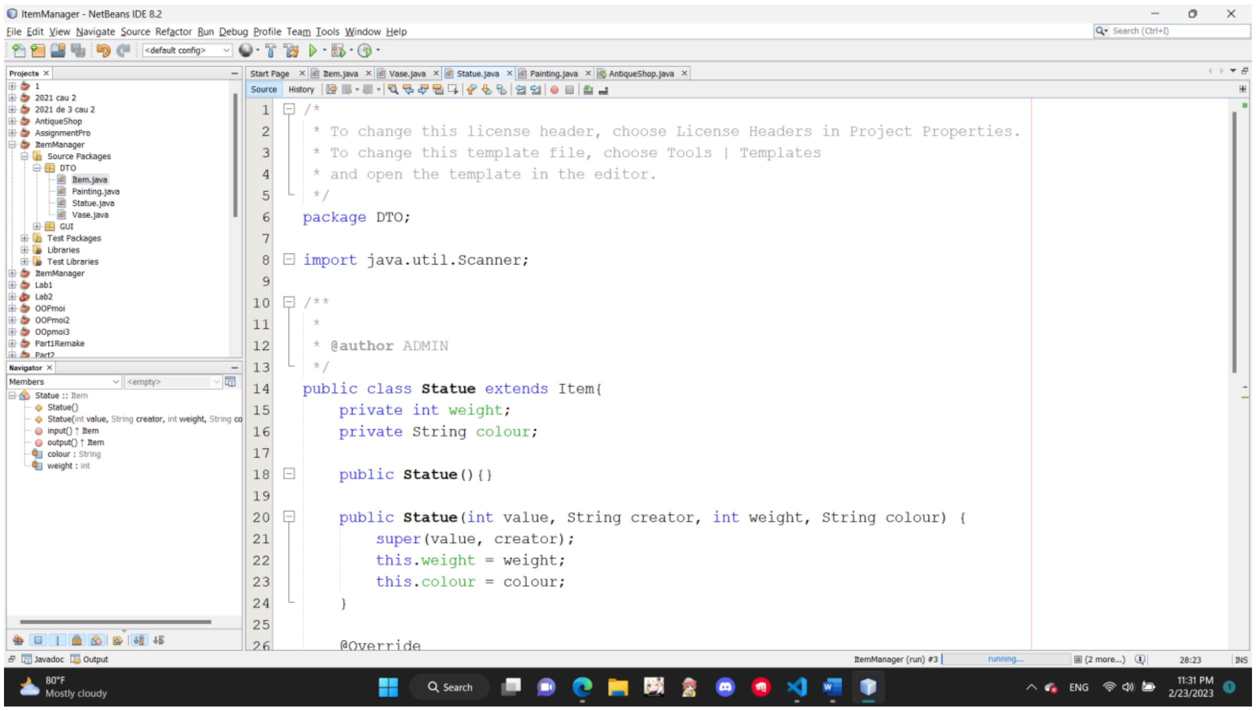


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package DTO;

import java.util.Scanner;

/**
 *
 * @author ADMIN
 */
public class Vase extends Item {
    private int height;
    private String material;

    public Vase(){

    }

    public Vase(int value, String creator, int height, String material ) {
        super(value, creator);
        this.height = height;
        this.material = material;
```

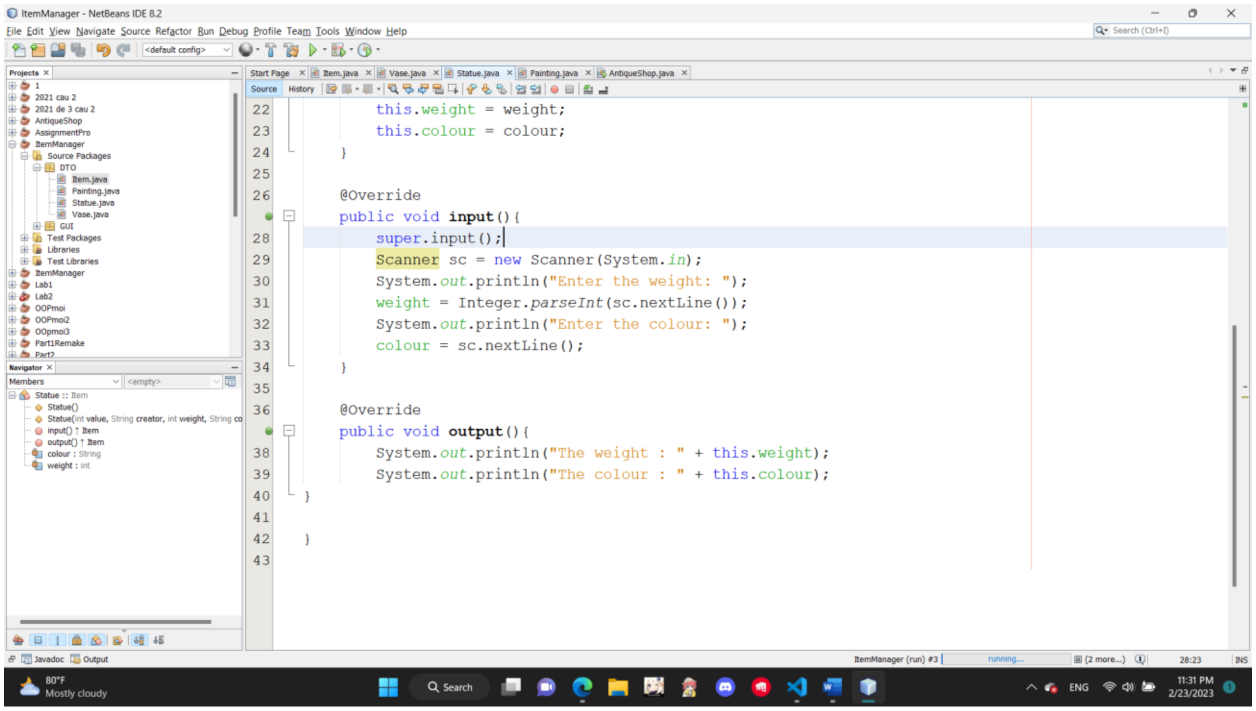


```java
        this.material = material;
    }

    public void input(){
        super.input();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the height: ");
        height = Integer.parseInt(sc.nextLine());
        System.out.println("Enter the material: ");
        material = sc.nextLine();
    }

    @Override
    public String toString() {
        return super.toString() + " - The height : " + this.height + "The material : " + this.material

//    public void output(){
//        super.output();
//        System.out.println(;
//}
    }
}
```

Statue:

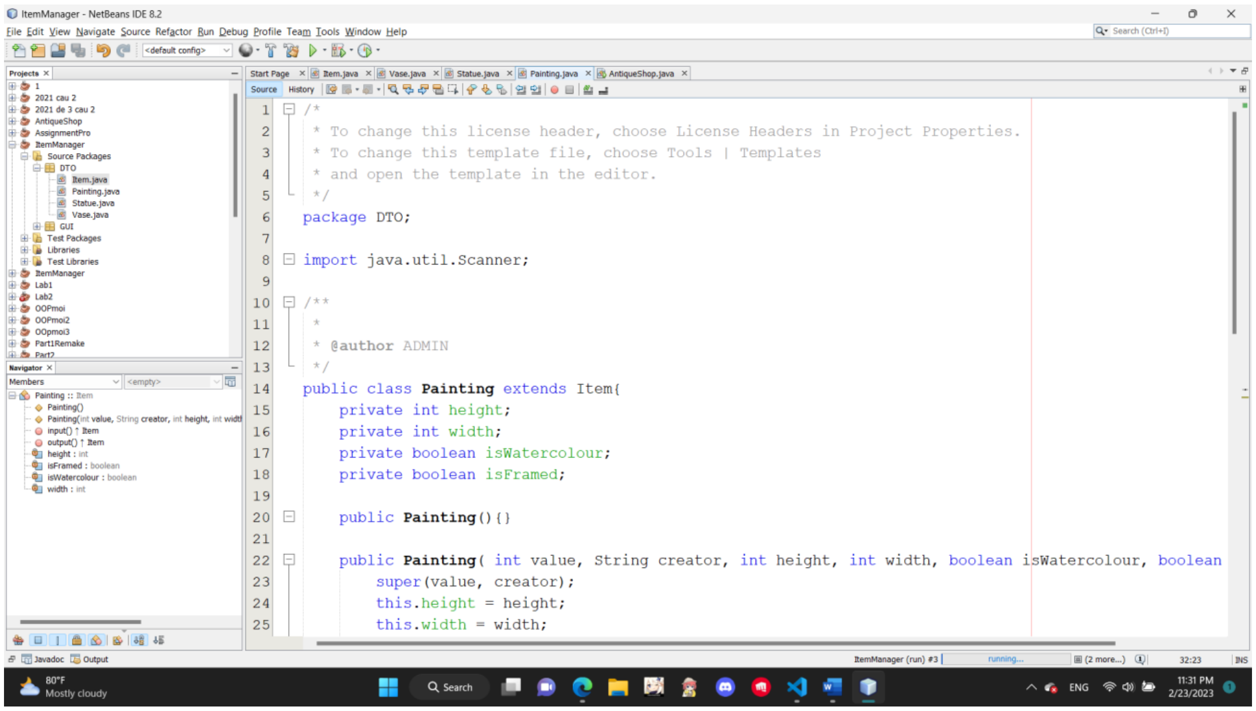


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package DTO;

import java.util.Scanner;

/**
 *
 * @author ADMIN
 */
public class Statue extends Item{
    private int weight;
    private String colour;

    public Statue(){}

    public Statue(int value, String creator, int weight, String colour) {
        super(value, creator);
        this.weight = weight;
        this.colour = colour;
    }

    @Override
```

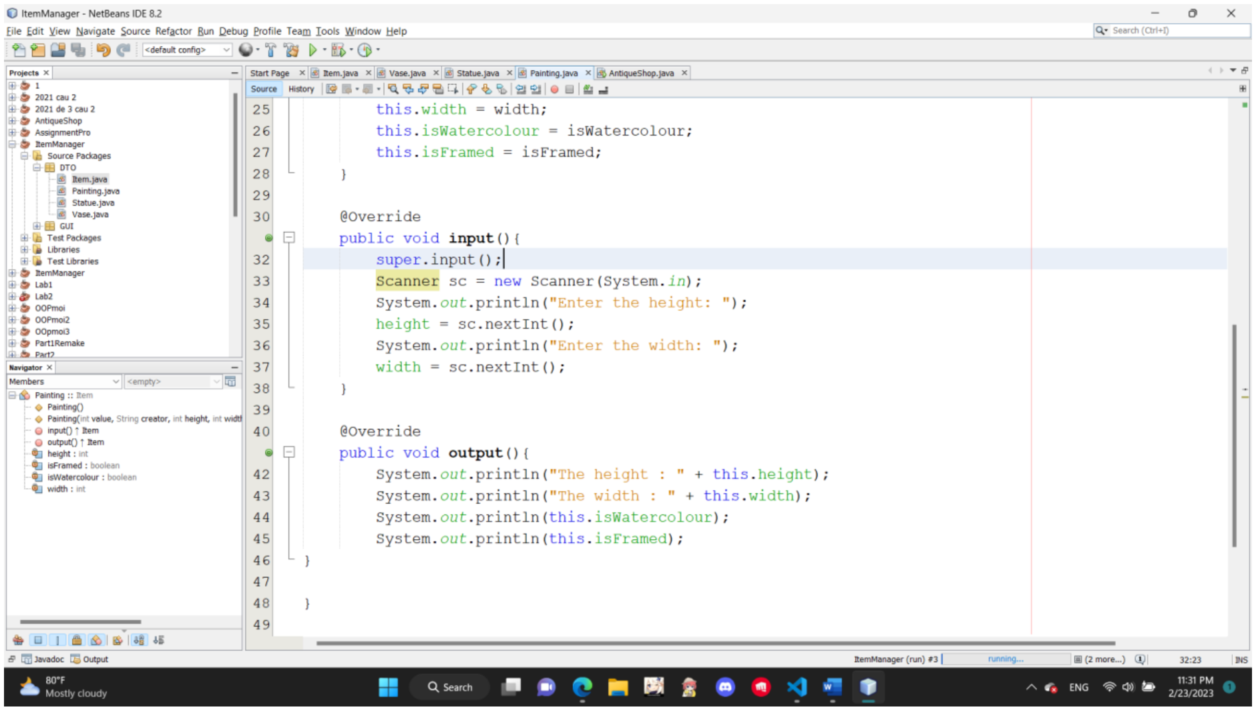


```java
        this.weight = weight;
        this.colour = colour;
    }

    @Override
    public void input(){
        super.input();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the weight: ");
        weight = Integer.parseInt(sc.nextLine());
        System.out.println("Enter the colour: ");
        colour = sc.nextLine();
    }

    @Override
    public void output(){
        System.out.println("The weight : " + this.weight);
        System.out.println("The colour : " + this.colour);
}

}

```

Painting:

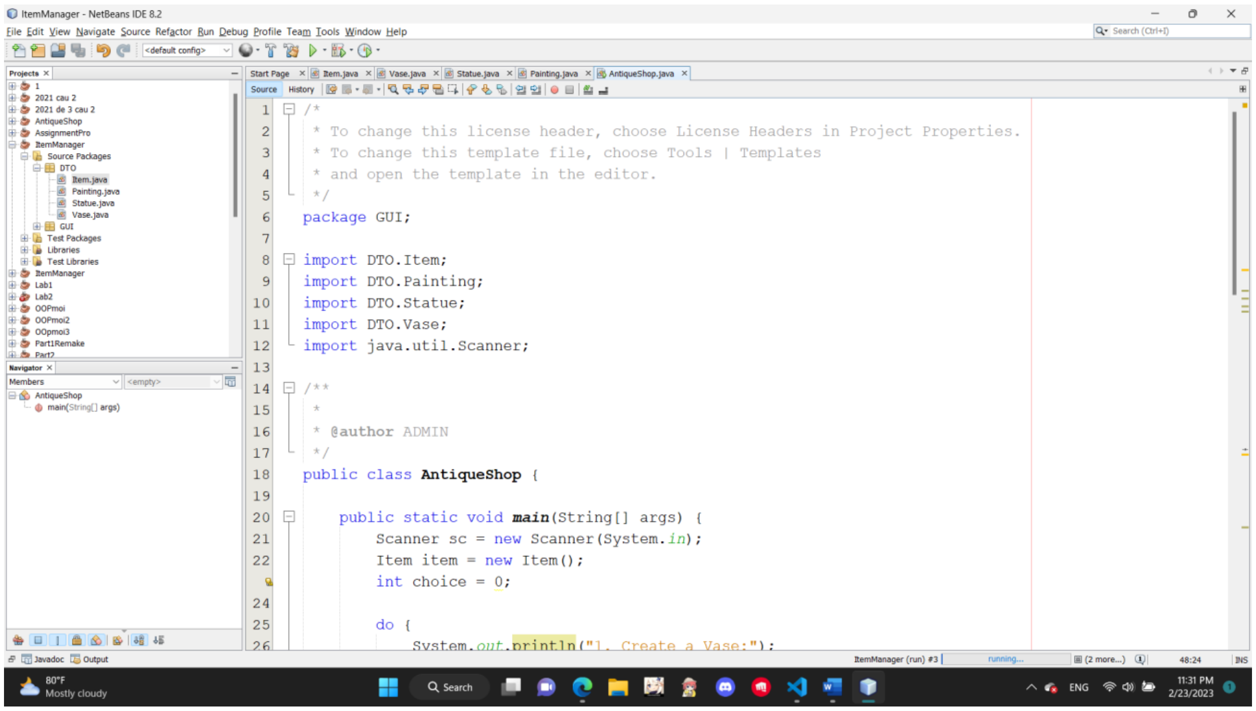


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package DTO;

import java.util.Scanner;

/**
 *
 * @author ADMIN
 */
public class Painting extends Item{
    private int height;
    private int width;
    private boolean isWatercolour;
    private boolean isFramed;

    public Painting(){}

    public Painting( int value, String creator, int height, int width, boolean isWatercolour, boolean
        super(value, creator);
        this.height = height;
        this.width = width;
```

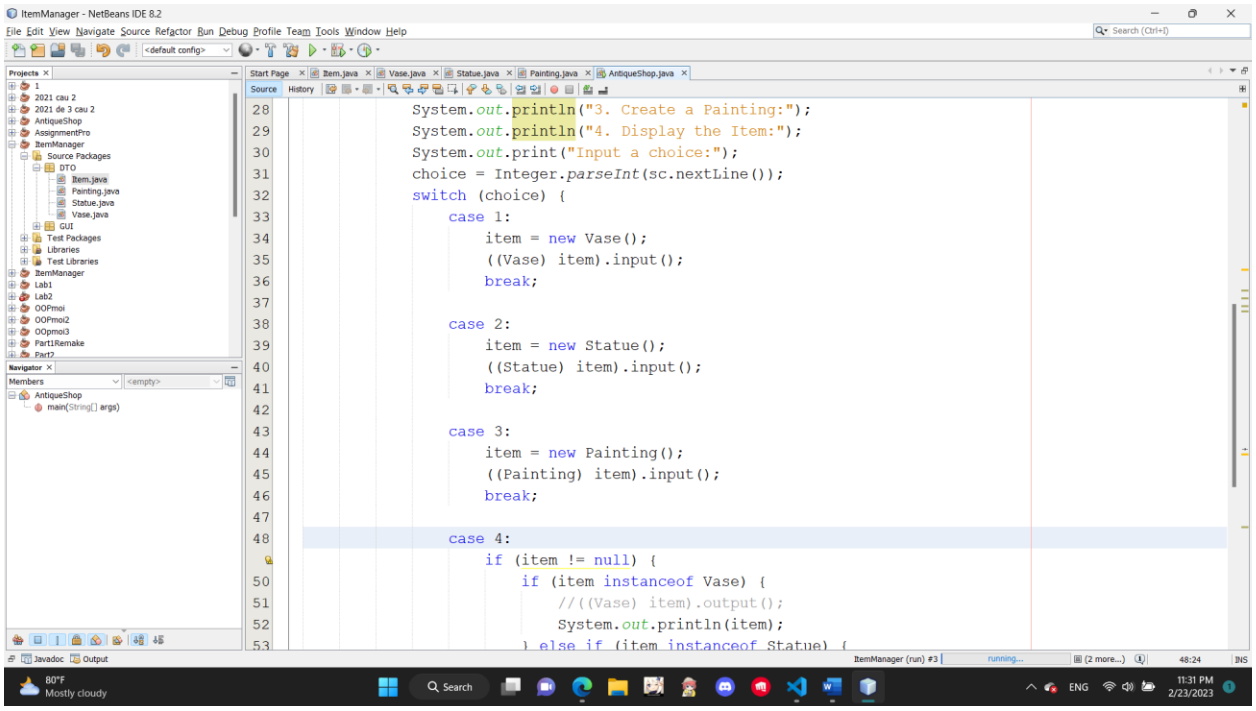


```java
        this.width = width;
        this.isWatercolour = isWatercolour;
        this.isFramed = isFramed;
    }

    @Override
    public void input(){
        super.input();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the height: ");
        height = sc.nextInt();
        System.out.println("Enter the width: ");
        width = sc.nextInt();
    }

    @Override
    public void output(){
        System.out.println("The height : " + this.height);
        System.out.println("The width : " + this.width);
        System.out.println(this.isWatercolour);
        System.out.println(this.isFramed);
}

    }

```

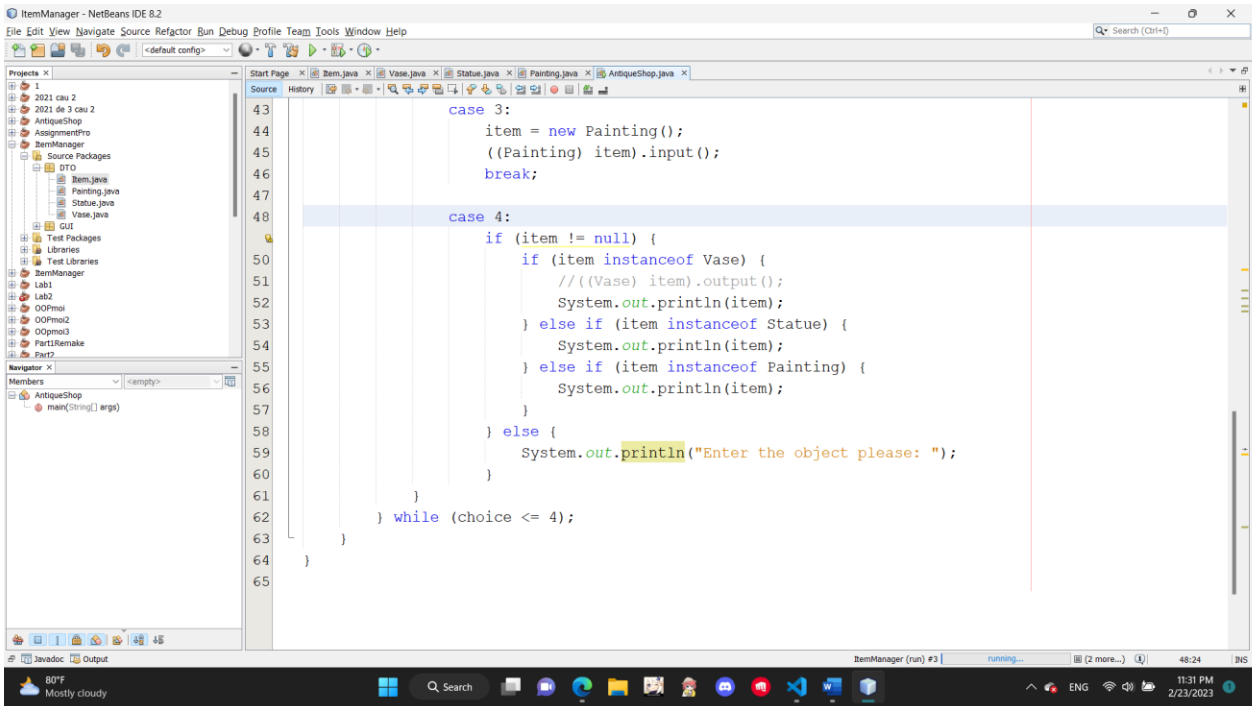
Antique:



```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package GUI;

import DTO.Item;
import DTO.Painting;
import DTO.Statue;
import DTO.Vase;
import java.util.Scanner;

/**
 *
 * @author ADMIN
 */
public class AntiqueShop {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Item item = new Item();
        int choice = 0;

        do {
            System.out.println("1. Create a Vase:");
```



```java
            System.out.println("3. Create a Painting:");
            System.out.println("4. Display the Item:");
            System.out.print("Input a choice:");
            choice = Integer.parseInt(sc.nextLine());
            switch (choice) {
                case 1:
                    item = new Vase();
                    ((Vase) item).input();
                    break;

                case 2:
                    item = new Statue();
                    ((Statue) item).input();
                    break;

                case 3:
                    item = new Painting();
                    ((Painting) item).input();
                    break;

                case 4:
                    if (item != null) {
                        if (item instanceof Vase) {
                            //((Vase) item).output();
                            System.out.println(item);
                        } else if (item instanceof Statue) {
```



```
                case 3:
                    item = new Painting();
                    ((Painting) item).input();
                    break;

                case 4:
                    if (item != null) {
                        if (item instanceof Vase) {
                            //((Vase) item).output();
                            System.out.println(item);
                        } else if (item instanceof Statue) {
                            System.out.println(item);
                        } else if (item instanceof Painting) {
                            System.out.println(item);
                        }
                    } else {
                        System.out.println("Enter the object please: ");
                    }
            }
        } while (choice <= 4);
    }
}
```

Part2:

What are objects in the program?

-item

What is the item variable storing?

-value and creator

Why must you cast to call the method inputVase()/outputVase()?

Import scanner

Super.

Object and class

What is the error thrown when you cast it wrong?

-Exception error

What methods can you call if you don't cast the item variable?

-toString

What is stored in the static heap, stack, dynamic heap?

A static variable is basically a global variable, even if you cannot access it globally. Usually there is an address for it that is in the executable itself. There is only one copy for the entire program. No matter how many times you go into a function call (or class) (and in how many threads!) the variable is referring to the same memory location.
The heap is a bunch of memory that can be used dynamically. If you want 4kb for an object then the dynamic allocator will look through its list of free space in the heap, pick out a 4kb chunk, and give it to you. Generally, the dynamic memory allocator (malloc, new, et c.) starts at the end of memory and works backwards.
Explaining how a stack grows and shrinks is a bit outside the scope of this answer, but suffice to say you always add and remove from the end only. Stacks usually start high and grow down to lower addresses. You run out of memory when the stack meets the dynamic allocator somewhere in the middle (but refer to physical versus virtual memory and fragmentation). Multiple threads will require multiple stacks (the process generally reserves a minimum size for the stack).